Optimal Reward Functions in Distributed Reinforcement Learning

David H. Wolpert and Kagan Tumer
NASA Ames Research Center, Mailstop 269-1, Moffett Field, CA 94035
{*dhw,kagan*}*@ptolemy.arc.nasa.gov*

We consider the problem of designing (perhaps massively) distributed collections of adaptive agents so as to optimize a world utility function dependent the behavior of the entire collection. We consider this problem when each agent's individual behavior is cast as striving to maximize an associated payoff utility function. The central issue in such design problems is how to initialize/update the payoff utility function so as to induce best possible world utility. Traditional "team game" approaches simply assign to each agent the world utility as its payoff utility function. In previous work we used the "Collective Intelligence" framework to derive a better choice of payoff utility functions, one that results in world utility performance up to orders of magnitude superior to that ensuing from use of the team game utility. In this paper we extend these results using a novel mathematical framework. We review the derivation under that framework of the general class of payoff utility functions that both are easy for the individual agents to learn and that, if learned well, result in high world utility. We then demonstrate experimentally that using these new utility functions can result in significantly improved performance over that of previously investigated collective intelligence payoff utilities, over and above those previous utilities' superiority to the conventional team game utility.

1 Introduction

In this paper we are interested in Multi-Agent Systems (MAS's) [1,2,3,4] where there is a provided world utility function that rates the possible histories of the full system. At the same time, each agent runs a reinforcement learning (RL) algorithm [5,6,7], to try to maximize its associated private utility function.

In such a system, we are confronted with an inverse problem: How should we initialize/update the agents' private utility functions to ensure that as the system unfolds the agents do not "work at cross-purposes", and their collective behavior maximizes the provided world utility function. Intuitively, to solve this inverse problem requires private utility functions that the agents can each learn well, but that also are "aligned" with the world utility. In particular, such alignment is necessary to avoid economics phenomena like the Tragedy of The Commons (TOC) [8] or Braess' paradox [9].

This problem is related to work in many other fields, including computational economics [10], mechanism design [11], reinforcement learning [7], statistical mechanics [12], computational ecologies [13], (partially observable) Markov decision processes [14] and game theory [11]. However none of these fields is both applicable in large, real-world problems, and also directly addresses the *gen-*

eral inverse problem rather than a very special instance of it. (In particular, the field of mechanism design is not generally applicable. A detailed discussion of related fields, involving hundreds of references is available [15].)

It's worth emphasizing that some of the previous work that does consider the general inverse problem does so by employing MAS's in which each agent uses RL [16,17]. However, in those cases, each agent generally receives the world utility function as its private utility function (i.e., implements a "team game" [18]). The shortcoming of such approaches, as expounded below and in previous work, is that they scale very poorly to large problems. (Intuitively, the difficulty is that each agent can have a hard time discerning the echo of its behavior on the world utility when the system is large.)

In previous work we modified these systems by using the COllective INtelligence (COIN) framework to to derive the alternative "Wonderful Life Utility" (WLU) [15], a private utility that generically avoids the pitfalls of the team game private utility [9,19,15,20]. For example, in some of that work we used the WLU as the private utility for distributed control of network packet routing [19]. Conventional approaches to packet routing have each router run a shortest path algorithm (SPA), i.e., each router routes its packets in the way that it expects will get those packets to their destinations most quickly. Unlike with a COIN, with SPA-based routing the routers have no concern for the possible deleterious side-effects of their routing decisions on the global goal (e.g., they have no concern for whether they induce bottlenecks). We ran simulations that demonstrated that a COIN-based routing system has substantially better throughputs than does the best possible SPA-based system [19], even though that SPA-based system has information denied the COIN system. In related work we have shown that use of the WLU automatically avoids the infamous Braess' paradox, in which adding new links can actually decrease throughput — a situation that readily ensnares SPA's.

As another example, we considered the pared-down problem domain of a congestion game [21], in particular a more challenging variant of Arthur's El Farol bar attendance problem [22], sometimes also known as the "minority game" [12]. In this problem, agents have to determine which night in the week to attend a bar. The problem is set up so that if either too few people attend (boring evening) or too many people attend (crowded evening), the total enjoyment of the attendees drops. Our goal is to design the reward functions of the attendees so that the total enjoyment across all nights is maximized. In this previous work we showed that use of the WLU can result in performance *orders of magnitude* superior to that of team game utilities.

In this article we extend this previous work, by investigating the impact of the choice of the single free parameter in the WLU (the "clamping parameter"),

which we simply set to 0 in our previous work. In particular, we employ some of the mathematics of COINs to determine the theoretically optimal value of the clamping parameter, and then present experimental tests to validate that choice of clamping parameter. In the next section we review the relevant concepts of COIN theory. Then we sketch how to use those concepts to derive the optimal clamping parameter. To facilitate comparison with previous work, we chose to conduct our experimental investigations of the performance with this optimal clamping parameter in variations of the Bar Problem. We present those variations in Section 3. Finally we present the results of the experiments in Section 4. Those results corroborate the predicted improvement in performance when using our theoretically derived clamping parameter. This extends the superiority of the COIN-based approach above conventional team-game approaches even further than had been done previously.

2 Theory of COINs

In this section we summarize that part of the mathematics of COINs that is relevant to the study in this article. We consider the state of the system across a set of consecutive time steps, $t \in \{0, 1, ...\}$. Without loss of generality, all relevant characteristics of agent η at time t — including its internal parameters at that time as well as its externally visible actions — are encapsulated by a Euclidean vector $\zeta_{\eta,t}$, the *state* of agent η at time t. $\zeta_{,t}$ is the set of the states of all agents at t, and ζ is the system's worldline, i.e., the state of all agents across all time.

World utility is $G(\zeta)$, and when η is an RL algorithm "striving to increase" its **private utility**, we write that utility as $\gamma_\eta(\zeta)$. (The mathematics can readily be generalized beyond such RL-based agents [15]. Here we restrict attention to utilities of the form $\sum_t R_t(\zeta_{,t})$ for **reward functions** R_t.

Definition 1: A system is **factored** if for each agent η individually,

$$\gamma_\eta(\zeta) \geq \gamma_\eta(\zeta') \quad \Leftrightarrow \quad G(\zeta) \geq G(\zeta') \ ,$$

for all pairs ζ and ζ that differ only for node η.

For a factored system, when every agents' private utility is optimized (given the other agents' behavior), world utility is at a critical point (e.g., a local maximum) [15]. In game-theoretic terms, optimal global behavior occurs when the agents' are at a private utility Nash equilibrium [11]. Accordingly, there can be no TOC for a factored system [15,19,20].) In addition, off of equilibrium, the private utilities in factored systems are "aligned" with the world utility.

Definition 2: The $(t = 0)$ **effect set** of node η at ζ, $S^{eff}_\eta(\zeta)$, is the set of all components $\zeta_{\eta',t'}$ for which the gradients $\vec{\nabla}_{\zeta_{\eta,0}}(\zeta)_{\eta',t'} \neq \vec{0}$. S^{eff}_η with no

specification of ζ is defined as $\cup_\zeta S_\eta^{eff}(\zeta)$. We will also find it useful to define $\hat{S}_\eta^{eff}$ as the set of all components that are not in S_η^{eff}. Intuitively, the $t = 0$ effect set of η is the set of all node-time pairs which, under the deterministic dynamics of the system, are affected by changes to η's $t = 0$ state.

Definition 3: Let σ be a set of agent-time pairs. $\mathrm{CL}_\sigma(\zeta)$ is ζ modified by "clamping" the states corresponding to the elements of σ to some arbitrary pre-fixed vector $\vec{\kappa}$. Then the (effect set) **Wonderful Life Utility** for node η (at time 0) is $WLU_\eta(\zeta) \equiv G(\zeta) - G(CL_{S_\eta^{eff}}(\zeta))$, where conventionally $\vec{\kappa} = \vec{0}$.

Note the crucial fact that to evaluate the WLU one does *not* need to know how to calculate the system's behavior under counter-factual starting conditions. All that is needed to evaluate WLU_η is the function $G(.)$, the actual ζ, and S_η^{eff} (which can often be well-approximated even with little knowledge about the system).

In previous work, we showed that effect set WLU is factored [20]. As another example, if $\gamma_\eta = G \; \forall \eta$ (a team game), then the system is factored. However for large systems where G sensitively depends on all components of the system, each agent may experience difficulty discerning the effects of its actions on G. As a consequence, each η may have difficulty achieving high γ_η in a team game. We can quantify this signal/noise effect by comparing the ramifications on $\gamma_\eta(\zeta)$ arising from changes to $\zeta_{\eta,0}$ with the ramifications arising from changes to $\zeta_{\hat{\eta},0}$, where $\hat{\eta}$ represents all nodes *other* than η. We call this quantification **learnability** [15]. A linear approximation to the learnability in the vicinity of ζ is the **differential learnability** $\lambda_{\eta,\gamma_\eta}(\zeta)$:

$$\lambda_{\eta,\gamma_\eta}(\zeta) \equiv \frac{\|\vec{\nabla}_{\zeta_{\eta,0}} \gamma_\eta(\zeta)\|}{\|\vec{\nabla}_{\zeta_{\hat{\eta},0}} \gamma_\eta(\zeta)\|} \; . \tag{1}$$

It can be proven that in many circumstances, especially in large problems, WLU has much higher differential learnability than does the team game choice of private utilities [15]. (Intuitively, this is due to the subtraction occurring in the WLU's removing a lot of the noise.) The result is that convergence to optimal G with WLU is much quicker (up to orders of magnitude so) than with a team game.

However the equivalence class of utilities that are factored for a particular G is not restricted to the associated team game utility and clamp-to-$\vec{0}$ WLU. Indeed, one can consider solving for the utility in that equivalence class that maximizes differential learnability. An approximation to this calculation is to solve for the factored utility that minimizes the expected value of $[\lambda_{\eta,WLR_\eta}]^{-2}$, where the expectation is over the values $\zeta_{,0}$.

A number of approximations have to be made to carry out this calcula-

tion[15]. The final result is that η should clamp to its empirical expected average action, where that average is over the elements in its training set[23]. Here, for simplicity, we do not actually make sure to clamp each η separately to its own average action, a process that involves η modifying what it clamps to in an online manner. Rather we clamp all agents to the same average action. We then made the guess that the typical probability distribution over actions is uniform. (Intuitively, we would expect such a choice to be more accurate at early times than at later times in which agents have "specialized".)

3 The Bar Problem

We focus on the following six more general variants of the bar problem investigated in our earlier work[20]: There are N agents, each picking one out of seven actions every week. Each action corresponds to attending the bar on some particular set of l out of the seven nights of the current week, where $l \in \{1, 2, 3, 4, 5, 6\}$.[a] At the end of the week the agents get their rewards and the process is repeated. For simplicity we chose the attendance profiles of each potential action so that when the actions are selected uniformly the resultant attendance profile across all seven nights is also uniform.

World utility is $G(\zeta) = \sum_t R_G(\zeta_{,t})$, where $R_G(\zeta_{,t}) \equiv \sum_{k=1}^{7} \phi(x_k(\zeta, t))$, $x_k(\zeta, t)$ is the total attendance on night k at week t, $\phi(y) \equiv y \exp(-y/c)$, and c is a real-valued parameter. (To keep the "congestion" level constant, for l going from 1 to 6, $c = \{3, 6, 8, 10, 12, 15\}$, respectively.) Our choice of $\phi(.)$ means that when either too few or too many agents attend some night in some week world reward R_G is low.

Since we are concentrating on the utilities rather than on the RL algorithms that use them, we use (very) simple RL algorithms. Each agent η has a 7-dimensional vector giving its estimates of the reward it would receive for taking each possible action. At the beginning of each week, each η picks the night to attend randomly, using a Boltzmann distribution over the seven components of η's estimated rewards vector. For simplicity, the temperature parameter of the Boltzmann distribution does not decay in time. However to reflect the fact that each agent operates in a non-stationary environment, reward estimates are formed using exponentially aged data: in any week t, the estimate η makes for the reward for attending night i is a weighted average of all the rewards it has previously received when it attended that night, with the weights given by

[a] In order to keep the learning difficulties faced by the agents similar for various choices of l, the agents always have seven action from which to choose. Each such action gets mapped to an "attendance" profile, e.g.., for $l = 2$, so that each agent must choose two nights, action one maps to attending on days one and two, action two maps to attending on days two and three etc.

an exponential function of how long ago each such reward was. To form the agents' initial training set, we had an initial period in which all actions by all agents were chosen uniformly randomly, before the learning algorithms were used to choose the actions.

4 Experimental Results

We investigate three choices of $\vec{\kappa}$: $\vec{0}$, $\vec{1} = (1,1,1,1,1,1,1)$, and the "average" action, $\vec{a} = \frac{\vec{l}}{7}$, where $l \in \{1,2,3,4,5,6\}$ depending on the problem. The associated WLU's are distinguished with a superscript. In the experiments reported here all agents have the same reward function, so from now on we drop the agent subscript from the private utilities. Writing them out, the three WLU reward functions are:

$$\begin{aligned}
R_{WL^{\vec{0}}}(\zeta_{,t}) &\equiv R_G(\zeta_{,t}) - R_G(CL^{\vec{0}}_{\eta}(\zeta_{,t})) \\
&= \phi_{d_\eta}(x_{d_\eta}(\zeta,t)) \; - \; \phi_{d_\eta}(x_{d_\eta}(\zeta,t) - 1) \\
R_{WL^{\vec{1}}}(\zeta_{,t}) &\equiv R_G(\zeta_{,t}) - R_G(CL^{\vec{1}}_{\eta}(\zeta_{,t})) \\
&= \sum_{d \neq d_\eta}^{7} \phi_d(x_d(\zeta,t)) \; - \; \phi_d(x_d(\zeta,t) + 1) \\
R_{WL^{\vec{a}}}(\zeta_{,t}) &\equiv R_G(\zeta_{,t}) - R_G(CL^{\vec{a}}_{\eta}(\zeta_{,t})) \\
&= \sum_{d \neq d_\eta}^{7} \phi_d(x_d(\zeta,t)) \; - \; \phi_d(x_d(\zeta,t) + a_d) \\
&+ \phi_{d_\eta}(x_{d_\eta}(\zeta,t)) \; - \; \phi_{d_\eta}(x_{d_\eta}(\zeta,t) - 1 + a_d)
\end{aligned}$$

where d_η is the night picked by η and $a_d = l/7$ The team game reward function is simply R_G. Note that to evaluate $R_{WL^{\vec{0}}}$ each agent only needs to know the total attendance on the night it attended. In contrast, R_G and $R_{WL^{\vec{a}}}$ require centralized communication concerning all 7 nights, and $R_{WL^{\vec{1}}}$ requires communication concerning 6 nights. Finally, note that when viewed in attendance space rather than action space, $CL^{\vec{a}}$ is clamping to the attendance vector $\vec{v}_i = \sum_{d=1}^{7} \frac{u_{d,i}}{7}$, where $u_{d,i}$ is the i'th component (0 or 1) of the the d'th action vector. So for example, for $l = 1$, $CL^{\vec{a}}$ clamps to $\vec{v}_i = \sum_{d=1}^{7} \frac{\delta_{d,i}}{7}$, where $\delta_{d,i}$ is the Kronecker delta function.

In the first experiment each agent had to select one night to attend the bar ($l = 1$). In this case, $\vec{\kappa} = \vec{0}$ is equivalent to the agent "staying at home," while $\vec{\kappa} = \vec{1}$ corresponds to the agent attending every night. Finally, $\vec{\kappa} = \vec{a} = \frac{\vec{1}}{7}$ is

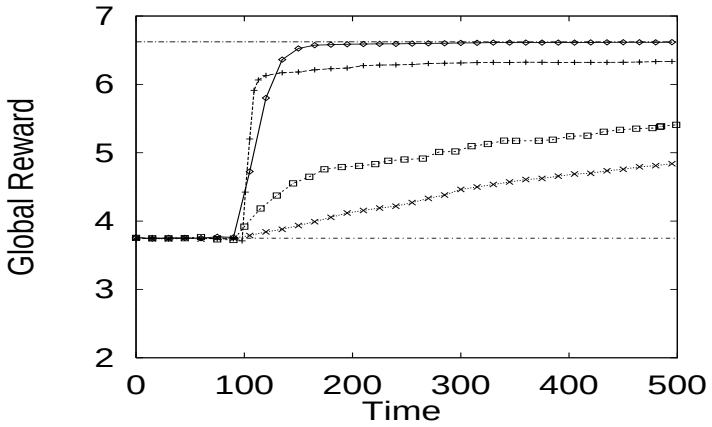

Figure 1: Reward function comparison when agents attend one night. ($WL_{\vec{a}}$ is $\diamond$; $WL_{\vec{0}}$ is $+$; $WL_{\vec{1}}$ is $\Box$; G is $\times$)

equivalent to the agents attending partially on all nights in proportions equivalent to the overall attendance profile of all agents across the initial training period. (Note, none of these "actions" are actually available to the agents. They simply use these fictional actions to compute their utilities, as described in Section 2.)

Figure 1 graphs world reward against time, averaged over 100 runs, for 60 agents and $c = 3$. (Throughout this paper, error bars are too small to depict.) The two straight lines correspond to the optimal performance, and the "baseline" performance given by uniform occupancies across all nights. Systems using $WL_{\vec{a}}$ and $WL_{\vec{0}}$ rapidly converged to optimal and to quite good performance, respectively. This indicates that for the bar problem the "mild assumptions" mentioned above hold, and that the approximations in the derivation of the optimal clamping parameter are valid.

Figure 2 shows how $t = 500$ performance scales with N for each of the reward signals. For comparison purposes the performance is normalized — for each utility U we plot $\frac{R_U - R_{base}}{R_{opt} - R_{base}}$, where R_{opt} and R_{base} are the optimal performance and a canonical baseline performance given by uniform attendance across all nights, respectively. Systems using R_G perform adequately when N is low. As N increases however, it becomes increasingly difficult for the agents to extract the information they need from R_G. Because of their superior learnability, systems using the WL rewards overcome this signal-to-noise problem to a great extent. Because the WL rewards are based on the *difference* between the actual state and the state where one agent is clamped, they are much less affected by the total number of agents. However, the action vector to which

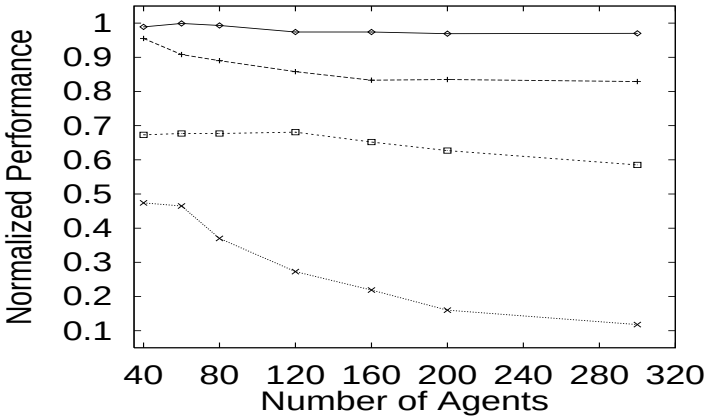

Figure 2: Scaling properties of the different reward function. ($WL_{\vec{a}}$ is $\diamond$; $WL_{\vec{0}}$ is $+$; $WL_{\vec{1}}$ is $\square$; G is $\times$)

agents are clamped also affects the scaling properties.

Figure 3 shows the normalized world reward obtained for the different private utilities as a function of l (i.e., when agents attend the bar on multiple nights in one week). $R_{WL^{\vec{a}}}$ performs well for all problems. $R_{WL^{\vec{1}}}$ on the other hand performs poorly when agents only attend on a few nights, but reaches the performance of $R_{WL^{\vec{a}}}$ when agents need to select six nights, a situation where the two clamping vectors are very similar ($\vec{1}$ and $\frac{\vec{6}}{7}$, respectively). $R_{WL^{\vec{0}}}$ shows a slight drop in performance when the number of nights to attend increases, while R_G shows a much more pronounced drop. Furthermore, in agreement with our previous results [20], despite being factored, the poor signal-to-noise in R_G results in poor performance with it for all problems. (Temperatures varied between .01 and .02 for the three WL rewards, and between .1 and .2 for the G reward, which provided the respective best performances for each.) These results confirm our theoretical prediction of what private utility converges fastest to the world utility maximum.

5 Conclusion

In this article we considered how to design large multi-agent systems to meet a pre-specified goal when each agent in the system uses reinforcement learning to choose its actions. We cast this problem as how to initialize/update the individual agents' private utility functions so that their collective behavior optimizes a pre-specified world utility function. The mathematics of COINs is specifically concerned with this problem. In previous experiments we showed

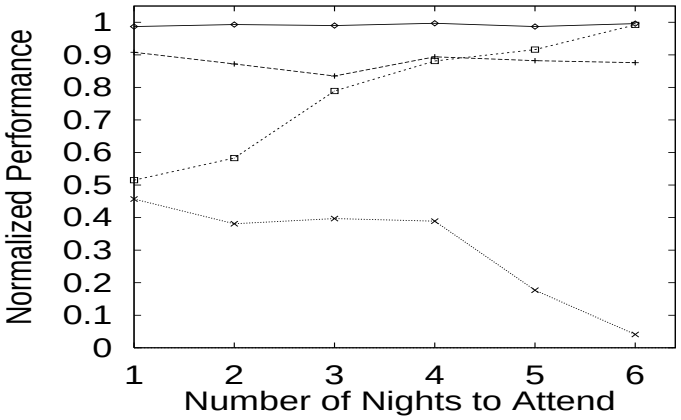

Figure 3: Behavior of different reward function with respect to number of nights to attend. ($WL^{\vec{a}}$ is $\diamond$; $WL^{\vec{0}}$ is $+$; $WL^{\vec{1}}$ is $\Box$; G is $\times$)

that systems based on that math far outperformed conventional "team game" systems, in which each agent has the world utility as its private utility function. Moreover, the gain in performance grows with the size of the system, typically reaching orders of magnitude for systems that consist of hundred of agents.

In those previous experiments the COIN-based private utilities had a free parameter, which we arbitrarily set to 0. However as synopsized in this paper, it turns out that a series of approximations in the allows one to derive an optimal value for that parameter. Here we have repeated some of our previous computer experiments, only using this new value for the parameter. These experiments confirm that with this new value the system converges to significantly superior world utility values, with less sensitivity to the parameters of the agents' RL algorithms. This makes even stronger the arguments for using a COIN-based system rather than a team-game system. Future work involves improving the approximations needed to calculate the optimal private utility parameter value. In particular, given that that value varies in time, we intend to investigate having it be calculated in an on-line manner.

1. C. Boutilier, Y. Shoham, and M. P. Wellman. Editorial: Economic principles of multi-agent systems. *Artificial Intelligence Journal*, 94:1–6, 1997.
2. J. M. Bradshaw, editor. *Software Agents*. MIT Press, 1997.
3. N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 1998.
4. K. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.
5. J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International*

Conference on Machine Learning, pages 242–250, June 1998.

6. L. P. Kaelbing, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
7. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
8. G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.
9. K. Tumer and D. H. Wolpert. Collective intelligence and Braess' paradox. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 104–109, Austin, TX, 2000.
10. T. Sandholm, K. Larson, M. Anderson, O. Shehory, and F. Tohme. Anytime coalition structure generation with worst case guarantees. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 46–53, 1998.
11. D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.
12. D. Challet and Y. C. Zhang. On the minority game: Analytical and numerical studies. *Physica A*, 256:514, 1998.
13. B. A. Huberman and T. Hogg. The behavior of computational ecologies. In *The Ecology of Computation*, pages 77–115. North-Holland, 1988.
14. M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, 1994.
15. D. H. Wolpert and K. Tumer. An Introduction to Collective Intelligence. Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center, 1999. URL:`http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`. To appear in Handbook of Agent Technology, Ed. J. M. Bradshaw, AAAI/MIT Press.
16. C. Claus and C. Boutilier. The dynamics of reinforcement learning cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, Madison, WI, June 1998.
17. T. Sandholm and R. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995.
18. R. H. Crites and A. G. Barto. Improving elevator performance using reinforcement learning. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems - 8*, pages 1017–1023. MIT Press, 1996.
19. D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, pages 952–958. MIT Press, 1999.
20. D. H. Wolpert, K. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, 49(6), March 2000.
21. D. H. Wolpert, K. Wheeler, and K. Tumer. General principles of learning-based multi-agent systems. In *Proceedings of the Third International Conference of Autonomous Agents*, pages 77–83, 1999.
22. W. B. Arthur. Complexity in economic theory: Inductive reasoning and bounded rationality. *The American Economic Review*, 84(2):406–411, 1994.
23. D. H. Wolpert. The mathematics of collective intelligence. pre-print, 2001.